Analytical Evaluation of Groupware Usability in Concerted Work Scenarios

António Ferreira
Pedro Antunes

DI-FCUL TR–05–15

July 2005

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749-016 Lisboa
Portugal

Analytical Evaluation of Groupware Usability in Concerted Work Scenarios

António Ferreira, Pedro Antunes

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749-016 Lisboa, Portugal

Abstract. This paper addresses the usability evaluation of groupware tools supporting teams working in an intensive concerted effort towards a shared goal. Considering the evaluation of this technology in such scenario is a complex and costly endeavor, we propose a discount analytical evaluation approach. Our approach uses available models of human performance to estimate groupware usability. The paper illustrates the use of the Keystroke-Level Model for analyzing a concerted work scenario and shows how the adopted evaluation method affords analytical experiments with alternative groupware designs. This paper contributes to understand the fine-grained details involved in groupware design.

1 Introduction and Motivation

Collaborative technologies place many challenges to usability evaluation, motivated by the number of users necessary to participate in the evaluation processes and the required control over technological factors and variables related to the group, task and context [7,5]. The complexity and cost associated to usability evaluation may be impeding the emergence of more successful groupware designs, highly usable and useful to individuals, work groups, and organizations.

Several discount methods have recently emerged with the purpose of reducing the complexity and cost of groupware usability evaluation. Many of them are adaptations of discount methods used with single-user software (singleware), such as groupware heuristic evaluation [2], groupware usability inspection [18], groupware walkthrough [17], and scenario based evaluation [6].

In this paper our research focus is on a particular type of groupware: the one that supports people working together in an intensive concerted effort towards a shared goal. Concerted work requires a significantly high level of workspace awareness, because individual actions affect the outcomes of the other members [14].

This specific type of groupware poses even more challenges to groupware usability evaluation, caused by the requirement to analyze the low-level details of individual and collaborative actions in shared workspaces, usually performed in very dynamic contexts. Furthermore, the impact of small design decisions on groupware support to concerted work scenarios is much higher than in other collaborative contexts, where the focus may be on more abstract activities, such as group decision making.

A set of analytical techniques have been developed in the Human-Computer Interaction field addressing the two concerns mentioned above: discount and attention to detail. These techniques rely on models of human performance to analyze usability problems and estimate task execution times of interactive tools. In this set we include the GOMS (Goals, Operations, Methods and Selection Rules) family of techniques [8], in particular the KLM (Keystroke-Level Model) [3,4].

These human performance models fall into the category of discount methods because they provide an analytic approach that can be applied without the participation of users and even without a prototype being developed [9]. Such models have been successfully used to benchmark many singleware design solutions [8].

As we show in the related work section, human performance models have mostly been used with singleware. In this paper we expand previous research on the possible benefits of using them with groupware [1].

We argue that human performance models contribute to groupware usability evaluation with additional insights about groupware design issues that are not covered by the other methods. The advantages of this approach emerge from the following fundamental characteristics of human performance models:

- Afford studying alternative design solutions in an analytical way [12,10]. This approach may save design time and effort by reducing the number of iterations and empirical tests necessary to revise and improve an initial design;
- Elucidate the assumed mechanisms and capabilities of the human processing system [3], which may be instrumental to develop more useable groupware tools;
- Specifically address situations where users accomplish tasks that they already master [8], disentangling the fine-grained details of concerted work;
- Offer quantitative estimates of human performance [3,8] which may be extrapolated to groupware interaction.

The paper is organized as follows. We start with a discussion of related work. Next, we describe a concerted work situation that will be the central case in our analysis. We proceed with the case analysis using the KLM approach. Then, we discuss possible design alternatives based on the insights offered by the case analysis. We finish the paper with a discussion of the benefits and limitations of our approach.

2 Related Work

Groupware walkthrough is a method adapting single-user cognitive walkthrough to the analytical evaluation of groupware [17]. It is based on the representation of collaborative activities using a set of mechanics of collaboration, i.e. fundamental types of collaborative interactions. Having this representation, a group of expert evaluators reviews and analyzes how the users' goals are supported. The major adaptations of cognitive walkthrough to the groupware context result from filtering out single-user actions and attaching the appropriate mechanics to typical collaborative tasks.

Another analytical method is groupware heuristic evaluation [2]. This method, adapted from single-user heuristic evaluation, relies on a small set of inspectors visually reviewing the compliance of a groupware tool with a list of heuristics. As with the groupware walkthrough approach, the list of heuristics is founded on the mechanics of collaboration.

Considering that both groupware walkthrough and groupware heuristic evaluation are dependent on the quality of the task analysis, another approach, called CUA (Collaboration Usability Analysis) appeared as an improved version of the mechanics of collaboration [16].

It is interesting to compare the CUA and human performance model approaches. Both analyze tasks using hierarchical decompositions but with significant differences in the intended level of detail. The CUA lowest granularity reduces collaboration tasks to the mechanics performed by users in shared workspaces, such as writing a message or obtaining a resource. Human performance models decompose tasks at a much lower level of detail; for instance, KLM analyses tasks at single keystrokes.

Single keystrokes are most times unrelated with collaborative work, notably when group decision making is involved, which is a strong argument in favor of high-level approaches such as CUA. However, we argue that going down to the keystroke level may provide additional insights about how users interact with groupware tools in concerted work situations. We provide two orders of reasons to support this argument:

– In concerted work, individual and group tasks are highly intertwined, so that individual tasks necessarily influence collaborative tasks and vice versa [14];
– Concerted work involves people performing repetitive and highly-mastered tasks, for which the human performance models have demonstrated good estimates [8].

We therefore hypothesize the design of collaborative tools for concerted work scenarios—where the designer may find necessary to optimize the effort applied by users in low-level tasks, even if only indirectly related with collaboration—may benefit from human performance analysis.

Nonetheless, the application of human performance models in the groupware context is very rare in the literature. DGOMS (Distributed GOMS) is an extension of GOMS to the group level of analysis [13]. The approach regards group work at a high level of detail, as a group task that can be successively decomposed in group subtasks until individual tasks can be identified. A new type of operator, called communication operator, is then defined to coordinate individual tasks executed in parallel. Therefore, this approach does not address concerted but coordinated work. As mentioned above, we focus on concerted work.

A similar approach is also suggested in a recent study of GOMS applied to a complex task executed by a team of users [11]. The task involved several users monitoring a display and executing actions in a coordinated way via a shared radio communication channel. As in the previous case, this does not address concerted work.

3 Case Description

The case explores a concerted work scenario involving a team with two members, Sophie and Charles, who work in different places. Sophie is highly trained in drawing vertical connection lines, while Charles is an expert in drawing horizontal connection lines. Given a board filled with points, the team has to quickly draw connections between all adjacent points using a groupware tool, as illustrated in Fig. 1. The board is characterized by a square arrangement of contiguous cells, numbered 1 to 9 in the example in Fig. 1, and by an initial state that always contains one horizontal and one vertical connection lines (these two lines are seeds for the forthcoming team activity).

The groupware tool has two workspaces for the following purposes: a) the public workspace displays a shared up-to-date view of the board; and b) the private workspace allows connecting points in a cell with horizontal or vertical lines, depending on the expertise of the team member. To simplify our analysis, we restrict the user interactions to the mouse and a single mouse button.

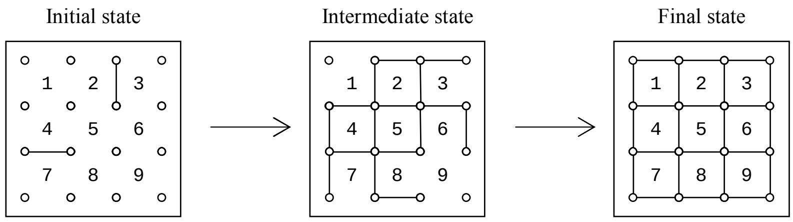

Fig. 1. The team draws horizontal and vertical lines, depending on the expertise of each member, to quickly connect all the points in the board

The rules for gradually connecting points in the board disallow manipulations in the public workspace. Instead, each member has to reserve points via the selection of the corresponding cell and by dragging it to the private workspace (see Fig. 2). Once there, the cell points can be connected in pairs, but only *if* at least one of the to-be-connected points is already linked to a third point in the same cell[1]. These modifications are made public when the cell is moved back to the public workspace.

Naturally, when a cell is dragged to a private workspace, the corresponding points are reserved (locked) in the public workspace (see exception in the next paragraph). To minimize inadvertent selections of reserved cells, the public workspace provides awareness by showing, next to the cell number, a letter that identifies the current owner, as depicted in Fig. 2.

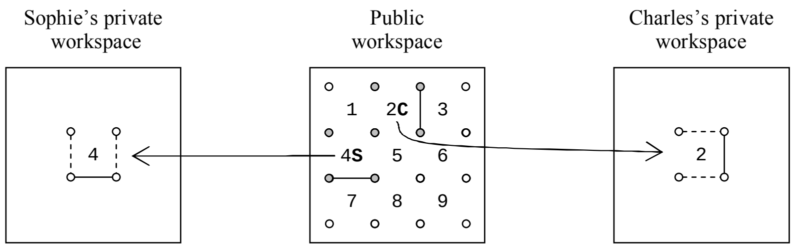

Fig. 2. The public workspace associates each reserved cell with the respective owner: Sophie's cell is identified with the letter `S` and Charles's with a `C`. Dashed lines represent work done in the private workspaces but not yet made public

The groupware tool also features automatic conflict resolution due to concurrent reservations of the same points. For example, if Sophie and Charles both select the same cell and simultaneously try to reserve it, then the groupware tool reserves the cell only to one of them, while the other is notified the cell is in use. A similar situation occurs when vertically and horizontally adjacent cells are reserved in parallel since the two points that belong to both cells cannot be in different private workspaces at the same time. The only *exception* to this rule is when a single point is shared between diagonally adjacent cells; in this case, the simultaneous reservation of such cells is allowed (see example in Fig. 2).

It is expected that the cells remain reserved for a relatively small amount of time due to the expertise of the team members and their eagerness to accomplish the task as fast as possible.

[1] For example, Sophie, who draws vertical connection lines, only selects and reserves cells that have at least one horizontal connection. Charles's behavior is analogous.

To demonstrate how this case represents concerted work, consider the intermediate state in Fig. 1. Now, suppose Sophie acts first by selecting and reserving cell 1; her action is visible in the public workspace due to the letter `S` in that cell. Based on this awareness information, Charles knows were Sophie is working and thus moves away from cell 1 and considers, for example, cells 6 or 7, which are available.

Furthermore, to quickly reach the final state (see Fig. 1) the team must be intensively working in harmony. The more horizontal connections exist, the more vertical connections can be drawn, and vice-versa. Conversely, if one member stops drawing connection lines, the other team member will soon also stop. For example, if Sophie arrives late to a situation where the board is still on the initial state, then Charles is capable of drawing only four horizontal connection lines (two each in cells 2 and 3), while being idle for the rest of the time. In other words, the actions of the team members are intertwined, this being a distinctive feature of concerted work [14].

4 Analytical Evaluation

The case analysis starts with a description of the entities and actions that characterize the concerted work environment, shown in Fig. 3.

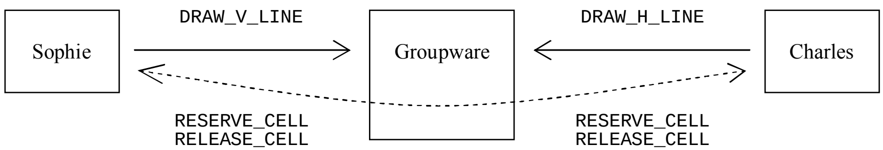

Fig. 3. Entities (boxes) and actions between entities (lines) characterizing the concerted work environment. The dashed line passing through the groupware tool means this entity is an intermediary for the actions (communication) among team members

There are three main entities in this case: Sophie and Charles are experts in drawing connection lines between adjacent points, although they have different specialties, which are, respectively, drawing vertical and horizontal connections. In addition, they are highly productive using the groupware tool, the third entity, which supports concerted work via private and public workspaces and workspace awareness as explained in the case description.

In this environment, team work results from a combination of individual and collaborative actions performed by the two members. Individual actions are `DRAW_V_LINE` and `DRAW_H_LINE`, for drawing one vertical/horizontal connection line between adjacent points in a cell. Being related with private workspaces they do not need any information to be delivered to the other team members. Collaborative actions are `RESERVE_CELL` and `RELEASE_CELL`, for moving a cell from the public to the private workspace, and vice-versa. These actions involve the public workspace and thus require the groupware tool to provide awareness information to all team members about their outcomes.

These two types of actions, supporting individual and collaborative work, are intertwined and under the control of the groupware tool, which means their design can influence individual, and especially, team performance.

The case analysis proceeds with a detailed specification of all the actions Sophie and Charles can perform using the adopted design constraints: mouse only inputs, exclusive reserves of points, drawing only in the private workspaces, and workspace awareness (see Table 1).

Table 1. Detailed descriptions of individual and collaborative actions in the work environment

Type	Action	Description
Individual	`DRAW_V_LINE`	Sophie (1) identifies a cell point, in her private workspace, that belongs to an horizontal connection but is missing a vertical connector; then she (2) presses the mouse button over the point and (3) moves the mouse cursor to the vertically adjacent point in the cell; once there, she (4) releases the mouse button
	`DRAW_H_LINE`	Charles (1) identifies a cell point, in his private workspace, that belongs to a vertical connection but is missing an horizontal connector; then he (2) presses the mouse button over the point and (3) moves the mouse cursor to the horizontally adjacent point in the cell; once there, he (4) releases the mouse button
Collaborative	`RESERVE_CELL`	The team member (1) identifies a candidate cell in the public workspace; then (2) presses the mouse button over the cell and (3) moves the mouse cursor to the private workspace; once there, (4) releases the mouse button
	`RELEASE_CELL`	The team member (1) identifies the cell in the private workspace; then (2) presses the mouse button over the cell and (3) moves the mouse cursor to the public workspace; once there, (4) releases the mouse button

The *candidate* cell mentioned in the `RESERVE_CELL` description is related to the interest of the team member in selecting the cell (see footnote 1). It also refers to a design feature addressing workspace awareness: letters, such as `S` and `C` (see Fig. 2), are used to make team members conscious about the cell availability and ownership. This awareness information is delivered after the `RESERVE_CELL` action is executed. Conversely, the `RELEASE_CELL` action updates the public workspace by removing the ownership letter from the cell and by making visible any new connections.

Our case analysis now proceeds with a groupware usability evaluation based on the KLM (Keystroke-Level Model) [3,4]. This model provides quantitative predictions of human performance based on the detailed descriptions of the actions in Table 1. In the KLM each action is converted into a sequence of mental and motor operators, shown in Table 2, whose individual execution times have been empirically established and validated in psychological experiences [3,15].

Table 2. Execution times (in milliseconds) for the KLM operators [3,15]

Operator	Execution time	Description
`M`	1200	Mental preparation (overall estimate)
`P`	1100	Point with mouse to target on a display (average distance)
`K`	100	Press or release mouse button (click takes 200 ms)

An important KLM requirement is modeling applies to expert error-free behavior only. This is met in our case since Sophie and Charles are highly trained in drawing line connectors and using the groupware tool.

To exemplify the conversion from the detailed textual description into a KLM representation, consider the `RELEASE_CELL` action in Table 1 and the illustration in Fig. 4.

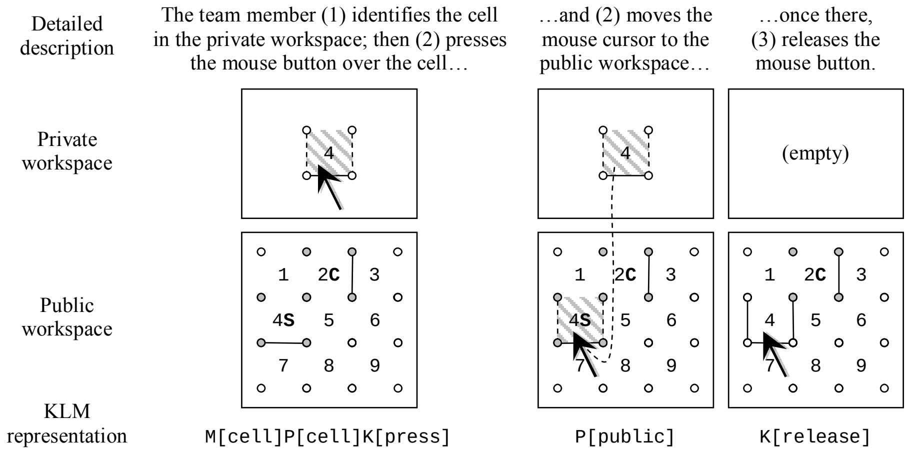

Fig. 4. Detailed description of the `RELEASE_CELL` action and corresponding KLM representation. The combined model is `M[cell]P[cell]K[press]P[public]K[release]`, with a predicted execution time of 1200+1100+100+1100+100 = 3600 ms

In step (1) of the `RELEASE_CELL` action the team member, say Sophie, identifies a, presumably worked, cell in her private workspace; this is converted into the `M[cell]` operator. Then she moves the mouse cursor over the cell, hence the `P[cell]`, and presses the mouse button, `K[press]`. In step (2) she moves the mouse cursor to the public workspace, an operation that requires a single `P[public]`, without a preceding `M[public]`, since there is no need for finding the workspace (it is always in the same place). In step (3) Sophie releases the mouse button, `K[release]`. The total predicted time for the execution of the `RELEASE_CELL` action, as for every other action, is obtained by adding the individual times of the KLM operators, as shown in Table 3.

Table 3. KLM representations and predicted execution times (in ms) for the individual and collaborative actions in the work environment (cf. detailed textual descriptions in Table 1)

Type	Action	KLM representation	Predicted times	Total
Individual	`DRAW_V_LINE` `DRAW_H_LINE`	(1) `M[point]`	1200	3600
		(2) `P[point]K[press]`	1100+100	
		(3) `P[next_point]`	1100	
		(4) `K[release]`	100	
Collaborative	`RESERVE_CELL`	(1) `M[cell]`	1200	3600
		(2) `P[cell]K[press]`	1100+100	
		(3) `P[private]`	1100	
		(4) `K[release]`	100	
	`RELEASE_CELL`	(1) `M[cell]`	1200	3600
		(2) `P[cell]K[press]`	1100+100	
		(3) `P[public]`	1100	
		(4) `K[release]`	100	

Interestingly, the KLM representations for the actions in our case are all essentially equal (a sequence of `MPKPK` operators), hence the predicted times are the same.

This suggests the required human skills for drawing a connection line between two points are very similar to those needed for moving a cell between workspaces, which seems intuitive. Furthermore, the predicted execution times also seem plausible if we consider Fitts's Law, the sizes of the objects, and the distances between them [3].

The estimates presented in Table 3 apply to single actions as if they were unrelated. To reveal the impact of collaborative actions in our case of intensive concerted work it is necessary to understand how work is carried out. We start with an analysis of individual behavior and then proceed with an assessment of team performance.

Given a candidate cell in the public workspace, each team member produces work by following one of the two possible sequences of actions shown in Table 4. Sequence `S1` is applicable, by either Sophie or Charles, to cell 1 in the intermediate state in Fig. 1; the sequence of actions `S2` is illustrated, for instance, in Fig. 2.

Table 4. Sequences of actions that team members can use to produce work, plus an assessment of the impact of collaborative and individual actions in total predicted times (in ms)

#	Sequence of actions	Total predicted time	Collaborative	Individual
`S1`	(1) `RESERVE_CELL` (2) `DRAW_LINE`[2] (3) `RELEASE_CELL`	3600 + 3600 + 3600 = 10800	7200 / 10800 = 66%	3600 / 10800 = 33%
`S2`	(1) `RESERVE_CELL` (2) `DRAW_LINE` (3) `DRAW_LINE` (4) `RELEASE_CELL`	3600 + 3600 + 3600 + 3600 = 14400	7200 / 14400 = 50%	7200 / 14400 = 50%

The data displayed in Table 4 is quite interesting, as it shows that collaborative actions, `RESERVE_CELL` and `RELEASE_CELL`, are more costly (7200 ms, 66% of total predicted time) than the individual action of drawing a connection line, `DRAW_LINE`, that characterizes sequence `S1`. It is therefore natural for the groupware designer to admit that team members will avoid such situation and instead prefer sequence `S2`, due to its lower collaboration overhead (50%) and small increase in execution time (33%) compared to twice the number of line drawings per reserved cell.

The advantages of sequence `S2` can also be taken into account in the design of the automatic conflict resolution mechanism, presented in the case description, so that overall team performance is optimized. When team members simultaneously try to reserve the same cell, or vertically or horizontally adjacent cells, the groupware tool could give preference, perhaps via heuristic rules, to the member that would be in condition of executing `S2`, in detriment of `S1`. For this to happen, however, the groupware tool would have to know the specialty of the team members, which, we find, is a reasonable use of context information.

Based on the analysis of individual behavior with the groupware tool, we can now evaluate team performance while doing intensive concerted work. Table 5 shows a simulation of the actions executed by Sophie and Charles starting from the initial state in Fig. 1 until they reach the final state (also depicted in the same figure).

[2] The `DRAW_LINE` action is an abstraction for `DRAW_V_LINE` and `DRAW_H_LINE`.

Table 5. Simulation of team activity for the scenario in Fig. 1 (time in ms). The letters `L`, `R`, `B`, and `T` are the initials for left, right, bottom, and top

Time	**Sophie**		**Charles**		**Observations**
0	S2	RESERVE_CELL(4)	S2	RESERVE_CELL(2)	(see initial state in Fig. 1)
3600		DRAW_V_LINE(L)		DRAW_H_LINE(T)	
7200		DRAW_V_LINE(R)		DRAW_H_LINE(B)	
10800		RELEASE_CELL(4)		RELEASE_CELL(2)	
14400	S2	RESERVE_CELL(7)	S2	RESERVE_CELL(3)	
18000		DRAW_V_LINE(L)		DRAW_H_LINE(T)	
21600		DRAW_V_LINE(R)		DRAW_H_LINE(B)	
25200		RELEASE_CELL(7)		RELEASE_CELL(3)	
28800	S2	RESERVE_CELL(6)	S2	RESERVE_CELL(8)	
32400		DRAW_V_LINE(L)		DRAW_H_LINE(T)	
36000		DRAW_V_LINE(R)		DRAW_H_LINE(B)	
39600		RELEASE_CELL(6)		RELEASE_CELL(8)	
43200	S1	RESERVE_CELL(2)	S1	RESERVE_CELL(4)	
46800		DRAW_V_LINE(L)		DRAW_H_LINE(T)	
50400		RELEASE_CELL(2)		RELEASE_CELL(4)	
54000	S1	RESERVE_CELL(1)		RESERVE_CELL(1)	*Reservation conflict*
57600		DRAW_V_LINE(L)	S1	RESERVE_CELL(6)	Charles sees cell 1 is reserved
61200		RELEASE_CELL(1)		DRAW_H_LINE(B)	
64800	S1	RESERVE_CELL(8)		RELEASE_CELL(6)	Sophie sees cell 3 is unavailable
68400		DRAW_V_LINE(R)	S1	RESERVE_CELL(1)	
72000		RELEASE_CELL(8)		DRAW_H_LINE(T)	
75600	S1	RESERVE_CELL(3)		RELEASE_CELL(1)	
79200		DRAW_V_LINE(R)	S1	RESERVE_CELL(7)	
82800		RELEASE_CELL(3)		DRAW_H_LINE(B)	
86400	S1	RESERVE_CELL(9)		RELEASE_CELL(7)	
90000		DRAW_V_LINE(R)		(WAITING)	Charles sees cell 9 is unavailable
93600		RELEASE_CELL(9)			
97200		(IDLE)	S1	RESERVE_CELL(9)	
100800				DRAW_H_LINE(B)	
104400				RELEASE_CELL(9)	
108000					(see final state in Fig. 1)

At the beginning of the simulation, Sophie and Charles apply the sequence of actions `S2` to cells 4 and 2, respectively. This takes them approximately 14400 milliseconds (see Table 4). They continue using this sequence until simulated time is 43200 ms; at this moment only S1 can be applied to cells 2 and 4. At time 54000 ms the team reaches the intermediate state shown in Fig. 1. Now, they both perceive cell 1 is available and simultaneously try to reserve it, which leads to a reservation conflict, as explained in the case description. Since both members can only use `S1` on cell 1, the groupware tool solves the conflict by arbitrarily granting the reservation to Sophie.

Charles notices his reservation was unsuccessful and moves on to cell 6. Around time 64800 ms, Sophie uses workspace awareness information to avoid cell 3 and, instead, reserve cell 6. At about 90000 ms, Charles also uses awareness information for not trying reserving cell 9, but in his case he has no other choice but wait for Sophie.

The simulation ends after 108 seconds of team (parallel) activity, this being the total predicted time for completing the task scenario in Fig. 1. Table 6 shows only about 36.6% of this time is used to draw connection lines, while 55% goes to collaborative actions. The rest of the time (8.3%) is "wasted" by the team members in waiting for a cell to become available or being idle with no further work to do.

Table 6. Summary of simulation results. The total simulated time, 108000 ms, is multiplied by the number of team members (Sophie and Charles) to obtain the % of simulation time

Type	Action	Executions	Predicted time	% of simulation time
Individual	`DRAW_V_LINE`	11	11 * 3600 = 39600	18.3%
	`DRAW_H_LINE`	11	11 * 3600 = 39600	18.3%
			Total = 79200	36.6%
Collaborative	`RESERVE_CELL`	8 + 9 = 17	17 * 3600 = 61200	28.3%
	`RELEASE_CELL`	8 + 8 = 16	16 * 3600 = 57600	26.6%
			Total = 136800	55.0%
Other	`(WAITING)`	2	2 * 3600 = 7200	3.3%
	`(IDLE)`	3	3 * 3600 = 10800	5.0%
			Total = 18000	8.3%

The results in Table 6, which we think are representative of a typical scenario of concerted work with this version of the groupware tool, provide a basis for making comparisons with other design options. This discussion will continue in the next section, where a design alternative will be evaluated using the same methodology.

5 Design Alternative and Discussion

Our design alternative for the groupware tool features a new type of awareness information and multiple cell reservations/releases. As before, reserved cells are marked with a letter that identifies the current owner, but now awareness information is also provided when a team member selects cells in the public workspace, by just clicking the mouse button over a cell, for example. The second feature allows multiple cells to be selected, and then reserved or released in a single step.

The reasons for these choices are twofold: first, as we will show, it is faster to select a cell in the public workspace than to reserve it, which means awareness information will be more up-to-date; second, the impact of collaborative actions in work production (see Table 4) can be reduced if the groupware tool allows multiple cells to be reserved or released at once, because more connection lines can be drawn consecutively in private workspaces.

The new features necessarily imply changes in the individual and collaborative actions that characterize the work environment. For example, the action `RESERVE_CELL` is now split in `SELECT_CELL_C` followed by `RESERVE_SELECTED`. Table 7 shows all the new textual descriptions.

Table 7. New descriptions of individual and collaborative actions in the work environment, including the corresponding KLM representations and total predicted time (in ms)

Type	Action	Description and KLM representation	Time
Individual	`DRAW_V_LINE` `DRAW_H_LINE`	Unchanged (see Table 1)	3600
		(1) `M[point]` (2) `P[point]K[press]` (3) `P[next_point]` (4) `K[release]`	
	`SELECT_CELL_I`	(similar to `SELECT_CELL_C`)	2500
	`SELECT_CELLS_I`	(similar to `SELECT_CELLS_C`)	4800
Collaborative	`SELECT_CELL_C`	The team member (1) identifies a candidate cell in the workspace and (2) clicks the mouse button over the cell	2500
		(1) `M[cell]` (2) `P[cell]K[press]K[release]`	
	`SELECT_CELLS_C`	The team member (1) identifies a candidate cell in the workspace and (2) presses the mouse button over the cell; then (3) identifies a second candidate cell that defines the desired imaginary rectangle, (4) moves the mouse cursor to the cell, and (5) releases the mouse button	4800
		(1) `M[cell]` (2) `P[cell]K[press]` (3) `M[next_cell]` (4) `P[next_cell]` (5) `K[release]`	
	`RESERVE_SELECTED`	The team member (1) presses the mouse button over a newly selected cell, (2) moves the mouse cursor to the private workspace, and (3) releases the mouse button	1300
		(1) `K[press]` (2) `P[private]` (3) `K[release]`	
	`RELEASE_SELECTED`	The team member (1) presses the mouse button over a newly selected cell, (2) moves the mouse cursor to the public workspace, and (3) releases the mouse button	1300
		(1) `K[press]` (2) `P[public]` (3) `K[release]`	

The difference between the `_I` and `_C` versions of `SELECT_CELL` and `SELECT_CELLS` is the workspace where the actions are executed: if the selection is made in a private workspace, then the `_I` (individual) version is used; the `_C` actions are used for selections in the public workspace, which produce awareness information to all team members, and thus are collaborative.

Table 7 shows the predicted time for `SELECT_CELL_C`, 2500 ms, is lower than the 3600 ms required for the older `RESERVE_CELL` action (see Table 3), which means team members should experience less time dealing with reservation conflicts. On the other hand, the time to reserve a single cell slightly increases because now it takes a `SELECT_CELL_C` followed by a `RESERVE_SELECTED`, with a total of 2500+1300 = 3800 ms, to perform what was previously done in `RESERVE_CELL` in 3600 ms. We consider this tradeoff acceptable because the time to recover from a reservation conflict is, at least, an order of magnitude greater than the extra 200 ms.

The actions `SELECT_CELLS_I` and `SELECT_CELLS_C` simplify the selection of multiple cells by allowing a team member to delineate an imaginary rectangle made of cells via the selection of two of its corners. This extra versatility, though, also increases the probability that two or more team members simultaneously select the same cells. To overcome this problem the groupware tool features an *anti*-selection mechanism that automatically excludes cells that are being selected by two or more members, as illustrated in Fig. 5.

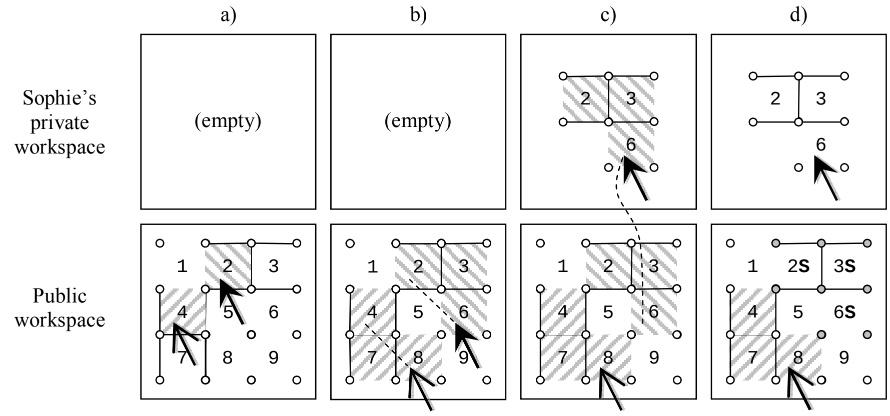

Fig. 5. Simultaneous selection and reservation of multiple cells: a) Sophie and Charles begin selecting multiple cells; b) the anti-selection mechanism prevents cell 5 from being selected; c) Sophie initiates a reservation by dragging the selected cells to her private workspace; and d) the multiple cell reservation is complete

The analysis of the design alternative now proceeds with a characterization of the sequences of actions that each team member can use to produce work (see Table 8).

As expected, if team members can only select single cells, they will still probably prefer reserving those candidate cells where two connection lines can be drawn, using sequence `S4`, in detriment of `S3`. This is because in `S4` the overhead of collaborative actions is lower than in `S3` (cf. similar situation in Table 4). However, as the data in Table 8 shows, if team members see an opportunity for reserving multiple candidate cells at once, then they will likely use sequence `S5` when *at least* three line drawings (`n` = 3) are doable in those cells. In these circumstances, the impact of collaborative actions is about 32%, or lower, this being unmatched by any of the sequences `S3` and `S4`. It is interesting to note that this design for cell selections affords a very clear and smooth definition of when to apply each sequence: if `n` ≥ 3 then use `S5`; else, if `n` = 2 then use `S4`; else, use `S3`.

Table 8. New sequences of actions that team members can use to produce work (times in ms)

#	Sequence of actions	Total predicted time	Collaborative	Individual
`S3`	(1) `SELECT_CELL_C`	2500 +	5100 / 11200 = 46%	6100 / 11200 = 54%
	(2) `RESERVE_SELECTED`	1300 +		
	(3) `DRAW_LINE`	3600 +		
	(4) `SELECT_CELL_I`	2500 +		
	(5) `RELEASE_SELECTED`	1300 = 11200		
`S4`	(1) `SELECT_CELL_C`	2500 +	5100 / 14800 = 34%	9700 / 14800 = 66%
	(2) `RESERVE_SELECTED`	1300 +		
	(3) `DRAW_LINE`	3600 +		
	(4) `DRAW_LINE`	3600 +		
	(5) `SELECT_CELL_I`	2500 +		
	(6) `RELEASE_SELECTED`	1300 = 14800		

S5	(1) SELECT_CELLS_C (2) RESERVE_SELECTED (3) DRAW_LINE . n (4) SELECT_CELLS_I (5) RELEASE_SELECTED	4800 + 1300 + 3600 . n + 4800 + 1300 = 3600 . n + 12200	7400 / total time n = 1 → 47% n = 2 → 38% n = 3 → 32% ...	(3600 . n + 4800) / total time n = 1 → 53% n = 2 → 62% n = 3 → 68% ...

We now present a simulation of team activity based on the analysis of individual behavior with the alternative groupware design (see Table 9). As before, Sophie and Charles start from the initial state in Fig. 1 until they reach the final state.

Table 9. New simulation of team activity (times in ms). The numbers in the arguments are cell identifiers. The observations are: a) see initial state in Fig. 1; b) cell 5 is automatically deselected; c) selection conflict; d) Charles sees cell 9 is unavailable; e) see final state in Fig. 1

Sophie			**Charles**			**Obser-vations**
Time	**S#**	**Action**	**Time**	**S#**	**Action**	
0	S5	SELECT_CELLS_C(4,7)	0	S5	SELECT_CELLS_C(2,3)	a)
4800		RESERVE_SELECTED	4800		RESERVE_SELECTED	
6100		DRAW_V_LINE(4,L)	6100		DRAW_H_LINE(2,T)	
9700		DRAW_V_LINE(4,R)	9700		DRAW_H_LINE(2,B)	
13300		DRAW_V_LINE(7,L)	13300		DRAW_H_LINE(3,T)	
16900		DRAW_V_LINE(7,R)	16900		DRAW_H_LINE(3,B)	
20500		SELECT_CELLS_I(4,7)	20500		SELECT_CELLS_I(2,3)	
25300		RELEASE_SELECTED	25300		RELEASE_SELECTED	
26600	S5	SELECT_CELLS_C(2,6)	26600	S5	SELECT_CELLS_C(4,8)	b)
31400		RESERVE_SELECTED	31400		RESERVE_SELECTED	
32700		DRAW_V_LINE(2,L)	32700		DRAW_H_LINE(4,T)	
36300		DRAW_V_LINE(3,R)	36300		DRAW_H_LINE(7,B)	
39900		DRAW_V_LINE(6,L)	39900		DRAW_H_LINE(8,T)	
43500		DRAW_V_LINE(6,R)	43500		DRAW_H_LINE(8,B)	
47100		SELECT_CELLS_I(2,6)	47100		SELECT_CELLS_I(4,8)	
51900		RELEASE_SELECTED	51900		RELEASE_SELECTED	
53200		SELECT_CELL_C(1)	53200		SELECT_CELL_C(1)	c)
55700	S3	SELECT_CELL_C(1)	55700	S3	SELECT_CELL_C(6)	
58200		RESERVE_SELECTED	58200		RESERVE_SELECTED	
59500		DRAW_V_LINE(1,L)	59500		DRAW_H_LINE(6,B)	
63100		SELECT_CELL_I(1)	63100		SELECT_CELL_I(6)	
65600		RELEASE_SELECTED	65600		RELEASE_SELECTED	
66900	S4	SELECT_CELL_C(9)	66900	S3	SELECT_CELL_C(1)	
69400		RESERVE_SELECTED	69400		RESERVE_SELECTED	
70700		DRAW_V_LINE(9,L)	70700		DRAW_H_LINE(1,T)	
74300		DRAW_V_LINE(9,R)	74300		SELECT_CELL_I(1)	
77900		SELECT_CELL_I(9)	76800		RELEASE_SELECTED	
80400		RELEASE_SELECTED	78100		WAITING(3600)	d)

81700		IDLE(11200)	81700	S3	SELECT_CELL_C(9)	
			84200		RESERVE_SELECTED	
			85500		DRAW_H_LINE(9,B)	
			89100		SELECT_CELL_I(9)	
			91600		RELEASE_SELECTED	
92900			92900			e)

The simulation shows the total predicted time for team activity is now 93 seconds, a decrease of almost 14% over the previous 108 seconds (see Table 6). We point out two main reasons for this improvement, both supported by data in Table 10: a) a reduction of the overhead of collaborative actions, 32.3% vs. 55.0%; and b) an increase of the proportion of line drawing actions, that effectively produce work, with respect to overall team activity, 42.6% vs. 36.6%.

Table 10. Summary of results for the new simulation.

Type	Action	Executions	Predicted time	% of simulation time
Individual	DRAW_V_LINE	11	11 * 3600 = 39600	21.3% vs. 18.3%
	DRAW_H_LINE	11	11 * 3600 = 39600	21.3% vs. 18.3%
	SELECT_CELL_I	2 + 3 = 5	5 * 2500 = 12500	6.7%
	SELECT_CELLS_I	2 + 2 = 4	4 * 4800 = 19200	10.3%
			Total = 110900	59.7% vs. 36.6%
Collaborative	SELECT_CELL_C	3 + 4 = 7	7 * 2500 = 17500	9.4%
	SELECT_CELLS_C	2 + 2 = 4	4 * 4800 = 19200	10.3%
	RESERVE_SELECTED	4 + 5 = 9	9 * 1300 = 11700	6.3%
	RELEASE_SELECTED	4 + 5 = 9	9 * 1300 = 11700	6.3%
			Total = 60100	32.3% vs. 55.0%
Other	(WAITING)	1	3600	1.9% vs. 3.3%
	(IDLE)	1	11200	6.0% vs. 5.0%
			Total = 14800	7.9% vs. 8.3%

Based on these analytical results, a designer would probably choose the design alternative as the starting point for the development of a prototype of the groupware tool and for testing with real teams.

6 Current Contributions and Future Work

We show in this paper how estimates drawn from research in cognitive engineering, which fundamentally address singleware, may be used to inform the design of groupware supporting intensive concerted work situations. As shown in the illustrated case, in concerted work individual and collaborative tasks are intertwined and mutually dependent. In such circumstances, groupware usability depends on very low-level details about how team members interact with the system, utilize workspace awareness to organize themselves, and set their work strategies by balancing costs associated to individual and collective actions. The proposed approach affords identifying and analyzing such tradeoffs, providing quantitative indications of which design alternatives may be beneficial to team work.

Research described in this paper is just a preliminary step in the direction of exploring models of human performance to estimate groupware usability. The estimates of human performance were based on experimental measures of time spent by humans executing single user operations. Experimental research with groupware will be accomplished in the near future. Also related with future work, we are investigating the development of operators specific to groupware interaction, based on the experience documented in this paper, analyzing typical (pattern-like) groupware mechanisms such as workspace awareness and floor control. Then, based on empirical tests, we will attempt to provide estimates for most common groupware interactions.

In future work, we will also explore how group dynamics influence overall groupware usability. For instance, regarding the example, the adoption of a more individualistic or more collaborative strategy may change according to time, as the group task evolves and fewer alternatives are available.

References

1. Antunes, P., Borges, M., Pino, J., Carriço, L.: Analyzing Groupware Design by Means of Usability Results. Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design (2005) 283–288
2. Baker, K., Greenberg, S., Gutwin, C.: Empirical Development of a Heuristic Evaluation Methodology for Shared Workspace Groupware. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (2002) 96–105
3. Card, S., Moran, T., Newel, A.: The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates (1983)
4. Card, S., Moran, T., Newell, A.: The Keystroke-Level Model for User Performance Time with Interactive Systems. Communications of the ACM, Vol. 23:7 (1980) 396–410
5. Fjermestad, J., Hiltz, S.: An Assessment of Group Support Systems Experimental Research: Methodology and Results. Journal of Management Information Systems, Vol. 15:3 (1999) 7–149
6. Haynes, S., Purao, S., Skattebo, A.: Situating Evaluation in Scenarios of Use. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (2004) 92–101
7. Ivory, M., Hearst, M.: The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys, Vol. 33:4 (2001) 470–516
8. John, B., Kieras, D.: Using GOMS for User Interface Design and Evaluation: Which Technique? ACM Transactions on Computer-Human Interaction, Vol. 3:4 (1996) 287–319
9. John, B.: Why GOMS? Interactions, Vol. 2:4 (1995) 80–89
10. Khalifa, M.: Computer-Assisted Evaluation of Interface Designs. The Data Base for Advances in Information Systems, Vol. 29:1 (1998) 66–881
11. Kieras, D., Santoro, T. Computational GOMS Modeling of a Complex Team Task: Lessons Learned. Proceedings of the 2004 Conference on Human Factors in Computing Systems (2004) 97–104
12. Kieras, D.: Model-Based Evaluation. In: Jacko, J., Sears, A. (eds.): The Human-Computer Interaction Handbook. Lawrence Erlbaum Associates (2003) 1139–1151
13. Min, D., Koo, S., Chung, Y., Kim, B.: Distributed GOMS: An Extension of GOMS to Group Task. Proceedings of the 1999 IEEE Conference on Systems, Man, and Cybernetics (1999) 720–725
14. Nunamaker, J., Briggs, R., Mittleman, D., Vogel, D., Balthazard, P.: Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings. Journal of Management Information Systems, Vol. 13:3 (1997) 163–207

15. Olson, J., Olson, G.: The Growth of Cognitive Modeling in Human-Computer Interaction. Human-Computer Interaction, Vol. 5. Lawrence Erlbaum Associates (1990) 221–265
16. Pinelle, D., Gutwin, C., Greenberg, S.: Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. ACM Transactions on Computer-Human Interaction (2003) 281–311
17. Pinelle, D., Gutwin, C.: Groupware Walkthrough: Adding Context to Groupware Usability Evaluation. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves (2002) 455–462
18. Steves, M., Morse, E., Gutwin, C., Greenberg, S.: A Comparison of Usage Evaluation and Inspection Methods for Assessing Groupware Usability. Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work (2001) 125–134